Semantics and Verification of Software
Lecture 18: Dataflow Analysis

Thomas Noll

Lehrstuhl für Informatik 2
RWTH Aachen University
noll@cs.rwth-aachen.de

http://www-i2.informatik.rwth-aachen.de/i2/svsw/

Summer semester 2007

Outline

1 Repetition: Solving Dataflow Equation Systems

2 Uniqueness of Solutions

3 Efficient Fixpoint Computation

4 The MOP Solution

Definition (Dataflow system)

A dataflow system $S = (Lab, E, F, (D, \sqsubseteq), \iota, \varphi)$ consists of

- a finite set of (program) labels Lab (here: Lab_c),
- a set of extremal labels $E \subseteq Lab$ (here: $\{\mathsf{init}(c)\}$ or $\mathsf{final}(c)$),
- a flow relation $F \subseteq Lab \times Lab$ (here: $\mathsf{flow}(c)$ or $\mathsf{flow}^R(c)$),
- a complete lattice $(D, \sqsubseteq)$ that satisfies ACC (with LUB operator $\bigsqcup$ and least element $\bot$),
- an extremal value $\iota \in D$ (for the extremal labels), and
- a collection of monotonic transfer functions $\{\varphi_l \mid l \in Lab\}$ of type $\varphi_l : D \to D$.

The Equation System

Definition (Dataflow equation system)

Let $S = (Lab, E, F, (D, \sqsubseteq), \iota, \varphi)$ be a dataflow system. S defines the following equation system over the set of variables $\{\mathsf{AI}_l \mid l \in Lab\}$:

$$\mathsf{AI}_l = \begin{cases} \iota & \text{if } l \in E \\ \bigsqcup \{\varphi_{l'}(\mathsf{AI}_{l'}) \mid (l', l) \in F\} & \text{otherwise} \end{cases}$$

The Functional and Its Fixpoint

Definition (Dataflow functional)

The equation system of a dataflow system $S = (Lab, E, F, (D, \sqsubseteq), \iota, \varphi)$ induces a *functional*

$$\Phi_S : D^n \to D^n : (d_{l_1}, \ldots, d_{l_n}) \mapsto (d'_{l_1}, \ldots, d'_{l_n})$$

where $Lab = \{l_1, \ldots, l_n\}$ and

$$d'_{l_i} := \begin{cases} \iota & \text{if } l \in E \\ \bigsqcup \{\varphi_{l'}(d_{l'}) \mid (l', l_i) \in F\} & \text{otherwise} \end{cases}$$

Corollary

The fixpoint of Φ_S is effectively computable by iteration:

$$\text{fix}(\Phi_S) = \bigsqcup \{\Phi_S^i(\bot_{D^n}) \mid i \in \mathbb{N}\}$$

where $\bot_{D^n} = (\bot_D, \ldots, \bot_D)$

Outline

Uniqueness of Solutions

Just as in the denotational semantics of `while` loops, solutions of dataflow equation systems are not unique.

Example 18.1

1. Available Expressions: see Exercise 9.2

2. Live Variables: consider

   ```
   while [x>1]¹ do
     [skip]²;
   [x := x+1]³;
   [y := 0]⁴
   ```

   $\implies LV_1 = LV_2 \cup (LV_3 \cup \{x\})$
   $LV_2 = LV_1 \cup \{x\}$
   $LV_3 = LV_4 \setminus \{y\}$
   $LV_4 = \emptyset$

   $\implies LV_3 = \emptyset$

   $\implies LV_1 = LV_2 \cup \{x\}$
   $\quad\quad = LV_1 \cup \{x\}$

   $\implies$ Solutions: $LV_1 = LV_2 = \{x\}$ or $\{x, y\}$, $LV_3 = LV_4 = \emptyset$

   Here: least solution $\{x\}$ (maximal potential for optimization)

Outline

A Worklist Algorithm I

Observation: fixpoint iteration computes every AI_l in every step
$\implies$ redundant if $\mathsf{AI}_{l'}$ at no F–predecessor l' changed
$\implies$ optimization by worklist

Algorithm 18.2 (Worklist algorithm)

$$
\begin{aligned}
&\text{Input:} && \textit{dataflow system } S = (Lab, E, F, (D, \sqsubseteq), \iota, \varphi) \\
&\text{Variables:} && W \in (Lab \times Lab)^*, \{\mathsf{AI}_l \in D \mid l \in Lab\} \\
&\text{Procedure:} && W := \varepsilon; \mathbf{for}\ (l, l') \in F\ \mathbf{do}\ W := (l, l') \cdot W;\ \textit{\% Initialize } W \\
&&& \mathbf{for}\ l \in Lab\ \mathbf{do} \qquad \textit{\% Initialize } \mathsf{AI} \\
&&& \quad \mathbf{if}\ l \in E\ \mathbf{then}\ \mathsf{AI}_l := \iota\ \mathbf{else}\ \mathsf{AI}_l := \bot_D; \\
&&& \mathbf{while}\ W \neq \varepsilon\ \mathbf{do} \\
&&& \quad (l, l') := \mathbf{head}(W); W := \mathbf{tail}(W); \\
&&& \quad \mathbf{if}\ \varphi_l(\mathsf{AI}_l) \not\sqsubseteq \mathsf{AI}_{l'}\ \mathbf{then} \qquad \textit{\% Fixpoint not yet reached} \\
&&& \qquad \mathsf{AI}_{l'} := \mathsf{AI}_{l'} \sqcup \varphi_l(\mathsf{AI}_l); \\
&&& \qquad \mathbf{for}\ (l', l'') \in F\ \mathbf{do}\ W := (l', l'') \cdot W; \\
&\text{Output:} && \{\mathsf{AI}_l \mid l \in Lab\}
\end{aligned}
$$

A Worklist Algorithm II

Example 18.3 (Worklist algorithm)

Available Expression analysis for $c = [\texttt{x := a+b}]^1$;

$\qquad\qquad\qquad\qquad\qquad\qquad [\texttt{y := a*b}]^2$;

$\qquad\qquad\qquad\qquad\qquad\qquad \texttt{while } [\texttt{y > a+b}]^3 \texttt{ do}$

$\qquad\qquad\qquad\qquad\qquad\qquad\quad [\texttt{a := a+1}]^4$;

$\qquad\qquad\qquad\qquad\qquad\qquad\quad [\texttt{x := a+b}]^5$

(cf. Examples 15.1 and 17.3)

Transfer functions: $\varphi_1(A) = A \cup \{\texttt{a+b}\}$

$\qquad\qquad\qquad\quad \varphi_2(A) = A \cup \{\texttt{a*b}\}$

$\qquad\qquad\qquad\quad \varphi_3(A) = A \cup \{\texttt{a+b}\}$

$\qquad\qquad\qquad\quad \varphi_4(A) = A \setminus \{\texttt{a+b}, \texttt{a*b}, \texttt{a+1}\}$

$\qquad\qquad\qquad\quad \varphi_5(A) = A \cup \{\texttt{a+b}\}$

Computation protocol: on the board

A Worklist Algorithm III

Properties of the algorithm:

Theorem 18.4 (Correctnes of worklist algorithm)

Given a dataflow system $S = (Lab, E, F, (D, \sqsubseteq), \iota, \varphi)$, Algorithm 18.2 always terminates and computes $\mathsf{fix}(\Phi_S)$.

Proof.

see [Nielson/Nielson/Hankin 2005, p. 75 ff]   □

Outline

The MOP Solution I

- Other solution method for dataflow systems
- MOP = Meet Over all Paths
- Analysis information for block B^l = least upper bound over all paths leading to l

Definition 18.5 (Paths)

Let $S = (Lab, E, F, (D, \sqsubseteq), \iota, \varphi)$ be a dataflow system. For every $l \in Lab$, the set of paths up to l is given by

$$Path(l) := \{[l_1, \ldots, l_{k-1}] \mid k \geq 1, l_1 \in E,$$
$$(l_i, l_{i+1}) \in F \text{ for every } 1 \leq i \leq k, l_k = l\}.$$

For a path $p = [l_1, \ldots, l_{k-1}] \in Path(l)$, we define the transfer function $\varphi_p : D \to D$ by

$$\varphi_p := \varphi_{l_{k-1}} \circ \ldots \circ \varphi_{l_1} \circ \mathsf{id}_D$$

(so that $\varphi_{[]} = \mathsf{id}_D$).

Definition 18.6 (MOP solution)

Let $S = (Lab, E, F, (D, \sqsubseteq), \iota, \varphi)$ be a dataflow system where
$Lab = \{l_1, \ldots, l_n\}$. The MOP solution for S is determined by

$$\mathsf{mop}(S) := (\mathsf{mop}(l_1), \ldots, \mathsf{mop}(l_n)) \in D^n$$

where, for every $l \in Lab$,

$$\mathsf{mop}(l) := \bigsqcup \{\varphi_p(\iota) \mid p \in Path(l)\}.$$

Remark:

- $Path(l)$ is generally infinite
$\Longrightarrow$ not clear how to compute $\mathsf{mop}(l)$
- In fact: MOP solution generally undecidable (later)

Example 18.7 (Live Variables; cf. Examples 15.4 and 17.4)

$c = [\texttt{x := 2}]^1;$
$\quad [\texttt{y := 4}]^2;$
$\quad [\texttt{x := 1}]^3;$
$\quad \texttt{if } [\texttt{y > 0}]^4 \texttt{ then}$
$\quad\quad [\texttt{z := x}]^5$
$\quad \texttt{else}$
$\quad\quad [\texttt{z := y*y}]^6;$
$\quad [\texttt{x := z}]^7$

$\implies Path(1) = \{[7,5,4,3,2],$
$\qquad\qquad\qquad [7,6,4,3,2]\}$

$\implies \mathsf{mop}(1) = \varphi_{[7,5,4,3,2]}(\iota) \sqcup \varphi_{[7,6,4,3,2]}(\iota)$
$= \varphi_2(\varphi_3(\varphi_4(\varphi_5(\varphi_7(\emptyset))))) \sqcup$
$\quad \varphi_2(\varphi_3(\varphi_4(\varphi_6(\varphi_7(\emptyset)))))$
$= \varphi_2(\varphi_3(\varphi_4(\varphi_5(\{\texttt{z}\})))) \sqcup$
$\quad \varphi_2(\varphi_3(\varphi_4(\varphi_6(\{\texttt{z}\}))))$
$= \varphi_2(\varphi_3(\varphi_4(\{\texttt{x}\}))) \sqcup$
$\quad \varphi_2(\varphi_3(\varphi_4(\{\texttt{y}\})))$
$= \varphi_2(\varphi_3(\{\texttt{x}, \texttt{y}\})) \sqcup \varphi_2(\varphi_3(\{\texttt{y}\}))$
$= \varphi_2(\{\texttt{y}\}) \sqcup \varphi_2(\{\texttt{y}\})$
$= \emptyset \sqcup \emptyset$
$= \emptyset$